Вопросы

1. **Что такое событие**

События – это функции, которые могут быть привязаны к элементам HTML страниц. Код событий выполнится только после того, как произойдет их активирующее действие. Разные типы событий имеют разные активирующие действия.

1. **Какие события вы знаете?**

**События мыши:**

* click – происходит, когда кликнули на элемент левой кнопкой мыши (на устройствах с сенсорными экранами оно происходит при касании).
* contextmenu – происходит, когда кликнули на элемент правой кнопкой мыши.
* mouseover / mouseout – когда мышь наводится на / покидает элемент.
* mousedown / mouseup – когда нажали / отжали кнопку мыши на элементе.
* mousemove – при движении мыши.

**События на элементах управления:**

* submit – пользователь отправил форму <form>.
* focus – пользователь фокусируется на элементе, например нажимает на <input>.

**Клавиатурные события:**

* keydown и keyup – когда пользователь нажимает / отпускает клавишу.

**События документа:**

* DOMContentLoaded – когда HTML загружен и обработан, DOM документа полностью построен и доступен.

**CSS events:**

* transitionend – когда CSS-анимация завершена.

1. **AddEventListerent**

Событию можно назначить *обработчик*, то есть функцию, которая сработает, как только событие произошло. Именно благодаря обработчикам JavaScript-код может реагировать на действия пользователя. Есть несколько способов назначить событию обработчик.

1. Использование атрибута onclick
2. AddEventListerent

Метод addEventListener позволяет добавлять несколько обработчиков на одно событие одного элемента.

element.addEventListener(event, handler, [options]);

|  |
| --- |
| event |
| Имя события, например "click". |
| handler |
| Ссылка на функцию-обработчик. |
| options |
| Дополнительный объект со свойствами. |

1. **Объект события**

Когда происходит событие, браузер создаёт объект events и записывает в него детали произошедшего события и передаёт его в качестве аргумента функции-обработчика.

Атрибуты событий:

* event.type – тип события
* event.target – объект, на котором сработал обработчик (могут попасть дочерние элементы)
* event.currentTarget – объект, на котором навешан обработчик
* event.clientX – координаты мыши в момент срабатывания обработчика по оси X
* event.clientY – координаты мыши в момент срабатывания обработчика по оси Y
* event – все детали события